UNITED STATES PATENT OFFICE

UTILITY PATENT APPLICATION

TITLE OF THE INVENTION:

COMPUTER CURVE CONSTRUCTION SYSTEM II

INVENTOR: Brigit Ananya, a Citizen of the United States, residing at 20725 Locust Drive Los

5   Gatos, CA 95033

RELATED APPLICATIONS

Provisional Application, No. 60/424,097, filed November 5, 2002, by Brigit Ananya for

"Computer Curve Construction System II".

BACKGROUND OF THE INVENTION

10   <u>Field of the Invention</u>

This invention relates generally to computer curve construction systems and methods. In

particular, it describes additional embodiments of a "Computer Curve Construction System"

described in U.S. Patent No. 6,441,823, *Ananya,* where the chosen curves are conical sections.

<u>Description of Related Art</u>

15   A conic is a mathematically defined curve. For example, as shown in Fig. 1, typically a

conic it is drawn (constructed) with a computer curve drawing program by setting a start point a_0,

an end point a_1, and a control point a_2, and then choosing any arbitrary point a_3 on the conic. The

tangent directions of the conic are determined by the control point a_2, and the shape of the conic

is determined by the chosen arbitrary point a_3. However, simply choosing an arbitrary point a_3

20   that will be somewhere on the conic does not allow a program operator to intuitively anticipate

the shape of the conic that will be constructed by the computer.

In contrast, U.S. Patent No. 6,441,823, *Ananya, B.*, describes a computer curve construction system for creating peak-point curves by selecting a star point, a start tangent direction, an end point spaced from the start point, an end tangent direction, and a peak point somewhere between the start and end points where the program calculates and draws a curve

5   connecting the start and end points passing through the peak point where the tangent of the curve at the peak point is parallel to the chord between the start and end points. [See Claims 13, 14, 15, &16, Col. 24 ll 7-54.] By defining a peak point, the curve construction system described in *Ananya* enables the operator to better anticipate, intuitively, the shape of the curve the program will construct responsive to operator input.

10   Conical section or conics have the advantage of being generally more geometric than Bezier curves. This geometric feature allows for easier operator conceptualization of a contemplated curve before inputting defining parameters. Secondly, behavior of conics are computationally predictable with established relationships that exist or can be easily defined. Yet for formulations and accurate numeric approximations by a computer (CPU) constructing the

15   conic, the defining signature constraints on the operator selected parameters must satisfy particular conditions before the computer program can allow construction of a desired curve.

SUMMARY OF THE INVENTION

An invented computer curve construction system is described that enables construction of conics more intuitively, predictably, and accurately. Also new constructions of particular cubic

20   Bezier curves and their conic analogs are described.

As in *Ananya* (supra) the newly invented computer curve constructions allow curves that consist of several curve components connected either with G^0 continuity, i.e. continuity of points (geometric continuity of order 0), G^1 continuity, i.e. continuity of points and tangent directions, just the angles of the tangents, (geometric continuity of order 1), or G^2 continuity, i.e. continuity

25   of points, tangent directions, and curvatures (geometric continuity of order 2).

In a first embodiment, construction of a conic peak-point curve is described where the operator inputs are:

(i)   the start point, a_0;

(ii)   the end point, a_1;

5   (iii)   the start tangent direction, e_0;

(iv)   the end tangent directions, e_1; and

(v)   the distance of the peak point, p from the chord between the start and end points.

The peak point is the point on the curve that is farthest away from the chord sometimes referred to as the shoulder point of the curve. For conics the peak point always lies on the centerline

10   segment connecting the center of the chord with the intersection point t of the rays extending in the start and end tangent directions e_0, e_1 respectively from the start and end points a_0, a_1 [hereinafter referred to as start and end tangents].

In a second embodiment, construction of a conic point-point curve is described where the operator inputs are:

15   (i)   the start point, a_0;

(ii)   the start tangent direction, e_0; and

(iii)   the peak point, p.

From these inputs, the computer system displays a guide area locating possible end point loci. The guide area is defined by 2 rays intersecting at a point s that lies on a one of the 2 rays that

20   extends from the start point a_0 in the direction of the peak point p at twice (2X) the distance of the peak point p from the start point a_0. The other ray extends from s in the direction opposite to the start tangent direction e_0.

(iv)   The operator then selects any point in the guide area as the end point a_1 of the curve.

The computer, using any suitable mathematical formulae then constructs a curve passing through the start point, a_0, peak point, p, and the end point a_1. The end point tangent direction e_1 is

5   derived from the known point of intersection of the start and end tangents coinciding with the intersection of start tangent and the line connecting the center of the chord between the start and end points a_0, a_1, and the peak point, p.

In a third embodiment, construction of a conic point-tangent curve is described wherein the operator inputs are:

10   (i)   the start point, a_0;

(ii)   the start tangent direction, e_0;

(iii)   the end point, a_1;

(iv)   the end point tangent directions, e_1; and

(v)   a fixed weight, w, or fixed cos-weight, v, for the curve.

15   The computer, using any suitable mathematical formulae, then constructs a curve passing through the start point, a_0, and the end point a_1. The input weight w is a parameter defined as the proportion between the distance of a peak point from the center point q of the chord between the start and end points a_0, a_1 and the distance of the peak point from the intersection point t of the start and end tangents. Accordingly, the input weight w chosen for the curve locates the peak

20   point on the centerline segment connecting q with t. In this construction system, a fixed cos-weight v is an arbitrarily defined positive parameter that utilizes a multiplication factor such as the trigonometric cosine relationship for computing a weight that limits the permitted range of the peak points of the possible curves. For example, when the angle α between the start and end

tangent directions, e_0, e_1, is set by operator input near 180°, the height of a fixed weight curve (the locus of the peak point p) approaches infinity as α approaches 180°, whereas the height of a fixed cos-weight curve is limited as α approaches 180°, e.g., if the fixed cos-weight v utilizes a factor $\cos(\alpha/2)$ for computing the weight w, which approaches zero as $\alpha \rightarrow 180°$, the loci of the

5   peak points for the possible curves approaches a limit.

In a fourth embodiment, construction of a conic point curve is described wherein the operator inputs are:

(i)   the start point, a_0;

(ii)   the start tangent direction, e_0; and

10   (iii)   the end point, a_1, and

(iv)   a fixed weight, w, or fixed cos-weight, v, for the curve

The computer, using any suitable mathematical formulae, then constructs a curve passing through the start point, a_0, and the end point a_1, where the end tangent direction e_1 is automatically set, e.g. in a symmetric way such that the angle between a ray extending from the

15   start point a_0 in the direction of the end point a_1, and the end tangent direction, e_1, equals the angle of between the start tangent direction, e_0 and that ray. As in the case of the conic point-tangent curve, the chosen weight w or cos-weight v locates the peak point, p on the centerline segment between the center point of the chord and the intersection point t of the start and end tangents.

20   There are essentially infinite possible conic point-tangent curve constructions, and, as well, infinite possible conic point curve constructions because of the possible weights, w, or cos-weights, v, that can be selected by an operator. The invented concept of using a fixed weight w or a fixed cos-weight v is also applicable to cubic Bezier curves and many other curve classes.

By choosing to place the peak point p on the centerline segment connecting the center of the chord between the start and end points with the intersection point of the start and end tangents, the selected weight w or cos-weight v determines the position of the peak point.

In a fifth embodiment, construction of a conic curvature curve is described where the operator inputs are:

(i)   the start point, a_0;

(ii)   the start tangent direction, e_0; whereupon

the computer displays a guideline perpendicular to the start tangent direction, e_0 for the center m_0 of the start curvature circle, r_0;

(iii)   the center m_0 of the start curvature circle r_0 on the displayed guideline; and

(iv)   the end point a_1; and

(v)   the end tangent direction, e_1.

For conic curvature curves, the start curvature determines the end curvature. The computer, using any suitable mathematical formulae, then establishes a center m_1 for the end curvature circle r_1 and draws a conic curve through the start and end points a_0, a_1, with respective start and end tangent directions of e_0, e_1 with the center m_0 of the start curvature circle r_0.

In a sixth embodiment, a construction system for a new class of point curvature curves including those for cubic Bezier curves and conics is described where the operator inputs are:

(i)   the start point, a_0;

(ii)   the start tangent direction, e_0; whereupon

the computer displays a guideline perpendicular to the start tangent direction, e_0 for the center m_0 of the start curvature circle r_0;

(iii)   the center m_0 of the start curvature circle r_0 on the displayed guideline; and

(vi)   the end point a_1.

The computer, using any suitable mathematical formulae, then constructs a curve passing through the start and end points a_0, a_1, with start tangent direction e_0 and center m_0 of the start curvature circle r_0. In this embodiment, like the previously described conic point curve construction embodiment, the end tangent direction e_1 is automatically set, e.g. in a symmetric way such that the angle between a ray extending from the start point a_0 in the direction of the end point a_1, and the end tangent direction, e_1, equals the angle of between the start tangent direction, e_0 and that ray. For conics, the center m_1 of the end curvature circle r_1 is automatically determined as shown for conic curvature curves. Since the end tangent direction is chosen in a symmetric way, the end curvature of the resulting conic equals the start curvature. For cubic Bezier curves, the center m_1 of the end curvature circle r_1 must be chosen by the program, e.g., in such a way that the end curvature equals the start curvature.

As described in *Ananya* (supra), it is possible to modify a curve by changing the position of a feature of any curve component while drawing it or after the drawing is completed. It is also possible to modify the type of continuity by which two curve components are connected, or to make two curves out of one curve, or one curve out of two curves, to delete and redraw any curve component, or to add or subtract curve components.

Last but not least it is possible to change the curve type. In particular, it is possible to transform a conic into one or several cubic Bezier curves (which are very close) and to transform a cubic Bezier curve into one or several conics (which are fairly close, but are chosen in such a way that allows for shape improvement).

This invented Computer Curve Construction System II enables the construction of conics more intuitively, predictably, and accurately, and it is also faster, more efficient and ergonomic for the operator.

BRIEF DESCRIPTION OF THE DRAWINGS

5 Fig. 1 is all the steps in constructing a conic the old way

Fig. 2 is two curves connected without curvature continuity

Fig. 3 is two curves connected with curvature continuity

Fig. 4 is a first step in constructing a conic peak-point curve.

Fig. 5 is a second step in constructing a conic peak-point curve.

10 Fig. 6 is a third step in constructing a conic peak-point curve.

Fig. 7 is a first step in constructing a conic point-point curve.

Fig. 8 is a second step in constructing a conic point-point curve.

Fig. 9 is a third step in constructing a conic point-point curve.

Fig. 10 is a first step in constructing a conic point-tangent curve.

15 Fig. 11 is a second step in constructing a conic point-tangent curve.

Fig. 12 is a first step in constructing a conic point curve.

Fig. 13 is a second step in constructing a conic point curve.

Fig. 14 is a first step in constructing a conic curvature curve.

Fig. 15 is a second step in constructing a conic curvature curve.

20 Fig. 16 is a third step in constructing a conic curvature curve.

Fig. 17 is a first step in constructing a conic point-curvature curve.

Fig. 18 is a second step in constructing a conic point-curvature curve.

Fig. 19 is a third step in constructing a conic point-curvature curve.

5   **DETAILED DESCRIPTION OF THE INVENTION**

<u>INTRODUCTION</u>

The present computer curve construction system and method is preferably implemented as part of a computer drawing program for drawing curves, which is a practical application in the industrial art of computer drawing. In one embodiment, the system may be implemented in any

10   suitable computer language for any operating system and any hardware platform. The system may be distributed through any medium, such as a disc, non-volatile memory, or being available for downloading on a network. In other embodiments the system may be implemented in any firmware or any hardware.

In Figs.1-19, all points are represented by small circles, and all tangent directions are

15   represented by lines with arrows. A tangent direction only indicates the direction or angle of a tangent vector; it does not indicate the length of the tangent vector.

The Computer Curve Construction System described in *Ananya* (supra) is incorporated by reference into this application to provide definition, context and meaning to the newly invented curve construction embodiments described below.

20   <u>FIGS. 4-6— CONIC PEAK-POINT CURVES</u>

As in *Ananya* (supra), in a first step shown in Fig. 4, a start point a_0 is set, and a start tangent direction e_0 is set. In a second step shown in Fig. 5, an end point a_1 is set and an end

tangent direction e_1 is set. Then a guide for the peak point is shown (the point on the curve at the greatest distance from the chord). The guide for the peak point is the centerline segment which connects the center of the chord with the intersection point of the start and end tangents. For a conic the peak always lies on the centerline segment. In a third step shown in Fig. 6, the peak point p is set anywhere on this guide. Since it is difficult to place p exactly on the guide, the point that is set is projected onto the guide to the point on the guide at the same distance from the chord. A conic curve is automatically drawn by the computer through points a_0 and a_1, with start and end tangent directions e_0 an e_1, and with the peak point p, according to any suitable set of mathematical formulas.

This conic curve is already drawn when the mouse button (input device) is pressed for the peak point. When the peak point is dragged by the input mechanism, (mouse) to a new location, and the conic curve is changed, the final shape of which is drawn when the input mechanism releases the peak point.

Additional curve components of any type may be constructed connected to the original curve with G^2, G^1, or G^0 continuity.

The conic peak-point curves are more intuitive, predictable, and accurate to construct than conics with typical computer drawing programs, because the peak point directly determines its path.

An exemplar set of formulas for determining a conic peak-point curve is as follows:

Let $q = 0.5(a_0 + a_1)$ be the center of the chord. The equation for the conic is

$$x(t) = \frac{(1-t)^2 b_0 + 2t(1-t)w b_1 + t^2 b_2}{(1-t)^2 + 2t(1-t)w + t^2},$$

where b_0, b_1, b_2 are the control points and w is the weight. The control point b_0 is the start point a_0, the control point b_1 is the intersection point of the start and end tangents, the control point b_2 is the end point a_1, and the weight w is the proportion between the distance of the peak point from q and the distance of the peak point from b_1.

5   The following assumptions are made:

Assume that $\text{sign}(e_0 \times (a_1 - a_0)) = \text{sign}(e_1 \times (a_0 - a_1)) = \text{sign}(e_0 \times e_1)$ (where "×" denotes the vector product between two 2-dimensional vectors) and that the peak point p lies on the guide described above.

The following formulas are provided;

10                            $b_0 = a_0$

$$b_1 = a_0 + \frac{(a_1 - a_0) \times e_1}{e_0 \times e_1} e_0$$

$b_2 = a_1$

$$w = \frac{|p_1 - q_1|}{|p_1 - b_1|}.$$

The parameter for the peak point of a conic is always 0.5, whereas for a cubic Bezier curve this parameter can have a different value.

15   The concept of peak-point curve shows very clearly why the conics have a better shape than the cubic Bezier curves. The reason is that the conics are more centered, since the peak point always lies on the centerline, whereas for cubic Bezier curves the peak point can lie elsewhere than on the centerline.

FIGS. 7-9 — CONIC POINT-POINT CURVES

As in *Ananya* (supra), in a first step shown in Fig. 7, a start point a_0, and a start tangent direction e_0 are set. In a second step shown in Fig. 8, a peak point p is set. Then a guide for the end point is shown. The guide uses the special point that lies on the ray starting at the start point in the direction of the peak point at twice the distance of the peak point from the start point. It is the area of the open convex polygon defined by the following two rays starting at this special point: the ray in the direction of the peak point from the start point and the ray in the opposite direction of the start tangent direction. In a third step shown in Fig. 9, the end point a_1 is set anywhere on this guide. A conic c is automatically drawn through points a_0 and a_1, with start tangent direction e_0 and peak point p, according to any suitable set of mathematical formulas. As shown in Fig. 9, for the example of conics (unlike the example of cubic Bezier curves), the end tangent direction e_1 is automatically determined, since the intersection point of the start and end tangents is determined. It equals the intersection point of the start tangent and the line connecting the center of the chord with the peak point.

This conic c is already drawn when the mouse button (input device) is pressed for the end point a_1, and when the mouse is dragged, the end point a_1 is dragged to a new position, and the conic c is changed, the final shape of which is drawn when the mouse is released.

Additional curve components of any type may be constructed to connect with G2, G1, or G0 continuity.

The conic point-point curves are easy to construct, because the end tangent direction of each curve component is automatically determined.

An exemplar set of formulas for determining a conic point-point curve is as follows:

The following assumption is made:

Assume that the end point a1 lies inside the guide described above.

The following formulas are provided:

Using the automatically determined end tangent direction e1, the formulas for a conic point-point curve are the same as the formulas for a conic peak-point curve.

5 ### FIGS. 10-11 — CONIC POINT-TANGENT CURVES

As in *Ananya* (supra), in a first step shown in Fig. 10, a start point a_0 and a start tangent direction e_0 are set. In a second step shown in Fig. 11, an end point a_1 and an end tangent direction e1 are set. A conic c is automatically drawn through start and end points a_0 and a_1, with start and end tangent directions e_0 and e_1, according to any suitable set of mathematical formulas.

10 The conic has a chosen weight w such as $w = \cos(\alpha/2)$, where α is the angle between e_0 and e_1. With this weight, if the angle between e_0 and the ray extending from the start point to the end point equals the angle between that ray and e_1, the conic is a circular arc. However, the user is allowed to choose any fixed weight or any fixed cos-weight. The cos-weight is a new concept that has not been used anywhere. If the weight is $(v)\cos(\alpha/2)$, we say that the cos-weight is v.

15 The advantage of a fixed cos-weight is, that if the angle α goes toward 180 degrees, the height of the curve is still restrained, whereas for a fixed weight the height stretches toward infinity. If the weight is 1, the conic is a parabola, which means that the conic point-tangent curve is the same as the Bezier point-tangent curve described in *Ananya*. (supra) However, there are infinitely more possibilities for conic point-tangent curves (and also for conic point curves as described

20 below), when choosing any weight or cos-weight.

These infinitely more possibilities for point-tangent and point curves can also be allowed for the example of cubic Bezier curves by choosing the peak point to lie on the center line segment with a chosen weight or cos-weight and computing the Bezier peak-point curve.

As shown in Fig. 11, the peak point p is automatically determined by the weight w. The proportion λ of the distance of the peak point from the center of the chord to the total length of the center line segment which is:

$$\lambda = \frac{w}{w + 1}$$

5   This shows where on the center line the peak point lies.

This conic c is already drawn when the mouse is dragged to the position of the end tangent direction e1, and when the mouse is dragged further, the end tangent direction e1 is dragged to a new position, and the conic c is changed, the final shape of which is drawn when the mouse is released.

10   Additional curve components of any type may be constructed to connect with G^2, G^1, or G^0 continuity.

The conic point-tangent curves are easy to construct, because the peak point is automatically determined.

An exemplar set of formulas for determining a conic peak-point curve is as follows:

15   The following assumptions are made:

Assume that $\text{sign}(e_0 \times (a_1 - a_0)) = \text{sign}(e_1 \times (a_0 - a_1)) = \text{sign}(e_0 \times e_1)$.

The following formulas are provided:

The weight w is the chosen weight or $(v)\cos(\alpha/2)$, where v is the chosen cos-weight,

The proportion λ of the distance of the peak point from the center of the chord to the total

20   length of the centerline segment is given by:

$$\lambda = \frac{w}{w + 1}$$

Everything else is determined in the same way as for the conic peak-point curves.

FIGS. 12-13 — CONIC POINT CURVES

As in *Ananya* (supra), in a first step shown in Fig. 12, a start point a_0 and a start tangent direction e_0 are set. In a second step shown in Fig. 13, an end point a_1 is set. A conic c is automatically drawn through start and end points a_0 and a_1 with start tangent direction e_0, according to any suitable set of mathematical formulas. As shown in Fig. 13, an end tangent direction e_1 for end point a_1 is automatically set, for example, in a symmetric way such that the angle between the chord and the end tangent direction equals the angle between the start tangent direction and the chord.

The conic has a chosen weight w (which can be determined by a chosen cos-weight v as described for conic point-tangent curves). As shown in Fig. 13, the peak point p is automatically determined by the weight w; this computation is done in the same way as for conic point-tangent curves.

In this particular embodiment the whole curve component c is symmetric with respect to the axis that is perpendicular to the chord and goes through the center of the chord.

This conic c is already drawn when the mouse button (input device) is pressed for the end point a_1, and when the mouse is dragged, the end point a_1 is dragged to a new position, and the conic c is changed, the final shape of which is drawn when the mouse is released.

Additional curve components of any type may be constructed to connect with G^2, G^1, or G^0 continuity.

The conic point curves are easy to construct, because the end tangent direction and the peak point are automatically determined.

An exemplar set of formulas for determining a conic point curve is as follows:

The following assumption is made:

Assume that the angle between e_0 and the ray extending from a_0 in the direction of a_1 is smaller than 90°:

The following formulas are provided:

After choosing the end tangent direction e_1 as described above, the formulas for the conic point curve are the same as the formulas for a the conic point-tangent curve.

FIGS. 14-16 — CONIC CURVATURE CURVES

As in *Ananya* (supra), in a first step shown in Fig. 14, a start point a_0 and a start tangent direction e_0 are set, and the guide for the center of the start curvature circle (the osculating circle of the start curvature) is shown. The guide is the line through a_0 perpendicular to e_0. In a second step shown in Fig. 15, a center m_0 of the start curvature circle r_0 is set. Since it is difficult to place m_0 exactly on the guide, the point that is set is projected perpendicularly onto the guide. In a third step shown in Fig. 16, an end point a_1 and an end tangent direction e_1 are set. For a conic, once the start curvature is chosen, the end curvature is automatically determined. As shown in Fig. 16, a center m_1 of the end curvature circle r_1 is drawn automatically. A conic c is automatically drawn through start and end points a_0 and a_1 with start and end tangent directions e_0 and e_1 and center m_0 of the start curvature circle r_0 according to any suitable set of mathematical formulas.

This conic c is drawn when the mouse button {input device] is dragged for the end tangent direction, and when the mouse is dragged further, the end tangent direction is dragged to a new location, and the conic c is changed, the final shape of which is drawn when the mouse is released.

Additional curve components of any type may be constructed to connect with G^2, G^1, or G^0 continuity.

The conic curvature curves are easy to construct, because the end curvature is automatically determined. So for all additional conic curvature curves that are connected with G^2 continuity, no curvature has to be drawn, whereas for Bezier curvature curves that are connected with G^2 continuity always the end curvature has to be drawn.

5   An exemplar set of formulas for determining a conic curvature curve is as follows:

The equation for the conic is

$$x(t) = \frac{(1-t)^2 b_0 + 2t(1-t)w b_1 + t^2 b_2}{(1-t)^2 + 2t(1-t)w + t^2},$$

where b_0, b_1, b_2 are the control points and w is the weight. The formulas for the start curvature k_0 and end curvature k_1 are:

$$k_0 = \frac{area(b_0, b_1, b_2)}{w^2 \, |b_1 - b_0|^3}$$

10

$$k_1 = \frac{area(b_0, b_1, b_2)}{w^2 \, |b_2 - b_1|^3}$$

where area($(b_0$, b_1, $b_2)$ is the area of the triangle formed by b_0, b_1, b_2.

The following assumption is made:

15   Assume that sign($e_0 \times (a_1 - a_0)$) = sign($e_1 \times (a_0 - a_1)$) = sign($e_0 \times e_1$) and that the end point lies on the same side of e_0 as the center of the start curvature circle.

The following formulas are provided:

$b_0 = a_0$

$$b_1 = a_0 + \frac{(a_1 - a_0) x e_1}{e_0 x e_1} e_0$$

20    $b_2 = a_1$

$$w = \sqrt{\frac{\text{area}(b_0, b_1, b_2)}{k_0 \, |b_1 - b_0|^3}}$$

$$k_1 = \frac{\text{area}(b_0, b_1, b_2)}{w^2 \, |b_2 - b_1|^3}$$

FIGS. 17-19 —POINT-CURVATURE CURVES

A sixth new embodiment of the Computer Curve Construction System II is for

5   constructing point-curvature curves. In a first step shown in Fig. 17, a start point a_0 and a start

tangent direction e_0 are set, and the guide for the center of the start curvature circle (the

osculating circle of the start curvature) is shown. The guide is the line through a_0 perpendicular

to e_0. In a second step shown in Fig. 18, a center m_0 of the start curvature circle r_0 is set. Since it

is difficult to place m_0 exactly on the guide, the point that is set is projected perpendicularly onto

10   the guide. In a third step shown in Fig. 19, an end point a_1 is set. A curve c is automatically

drawn through start and end points a_0 and a_1, with start tangent direction e_0 and center m_0 of the

start curvature circle r_0 according to any suitable set of mathematical formulas. As shown in Fig.

19, an end tangent direction e_1 for end point a_1 is automatically set, for example, in a symmetric

way such that the angle between the ray extending from a_0 to a_1 and the end tangent direction

15   equals the angle between the start tangent direction and that ray.

For the example of conics, the center m_1 of the end curvature circle r_1 is automatically

determined as shown for conic curvature curves. Since the end tangent direction is chosen in a

symmetric way, the end curvature of the resulting conic equals the start curvature.

For the example of cubic Bezier curves, the center m_1 of the end curvature circle r_1 is

20   chosen in a particular way, for example in such a way that the end curvature equals the start

curvature.

This curve c is already drawn when the mouse button (input device) is pressed for the end point, and when the mouse is dragged, the end point is dragged to a new location, and the curve c is changed, the final shape of which is drawn when the mouse is released.

Additional curve components of any type may be constructed to connect with G^2, G^1, or

5   G^0 continuity.

The point-curvature curves are easy to construct, because the end tangent and end curvature are automatically determined. So for all additional point-curvature curves that are connected with G^2 continuity, no curvature has to be drawn.

An exemplar set of formulas for determining the point-curvature curve for conics is as

10   follows:

The following assumption is made:

Assume that $sign(e_0 \times (a_1 - a_0)) = sign(e_1 \times (a_0 - a_1)) = sign(e_0 \times e_1)$ and that the end point lies on the same side of e_0 as the center of the start curvature circle.

After choosing the end tangent direction e_1 as described above, the formulas for the conic

15   point-curvature curve are the same as the formulas for the conic curvature curve.

An exemplar set of formulas for determining the point-curvature curve for cubic Bezier curves is as follows:

The following assumption is made:

Assume that $sign(e_0 \times (a_1 - a_0)) = sign(e_1 \times (a_0 - a_1))$ and that the end point lies on the

20   same side of e_0 as the center of the start curvature circle.

After choosing the end tangent direction e_1 and the center m_1 of the end curvature circle as described above, the formulas for the Bezier point-curvature curve are the same as the formulas for the Bezier curvature curve.

MODIFYING TYPES OF CURVE COMPONENTS

Besides modifying different types as described in the *Ananya* (supra) it is also possible to change between cubic Bezier curves and conics.

Change from a cubic Bezier curve to conics:

5   If the angle between the start and end tangents is smaller than 180 degrees, the resulting conic is the conic peak-point curve with the same start and end points and the same start and end tangent directions and the peak point on the center line at the same distance from the chord as the peak point of the cubic Bezier curve. If the angle between the start and end tangent is larger than or equal to 180 degrees, the cubic Bezier curve is first changed into two cubic Bezier curves

10   which connect at the peak point of the original cubic Bezier curve. Then each Bezier curve is changed into a conic after finding its peak point, which is the point where the tangent is parallel to the chord, that can be found by using the section "Computing Points and Tangent Directions of a Cubic Bezier Curve" described in *Ananya* (supra).

Change from a conic to cubic Bezier curves:

15   If the angle between the start and end tangents is smaller than 90 degrees, the resulting cubic Bezier curve is the Bezier peak-point curve with the same start and end points and the same start and end tangent directions and the same peak point. If the angle between the start and end tangents is larger than or equal to 90 degrees (it is still smaller than 180 degrees because of the assumptions for conics), the conic is first changed into two conics which connect at the peak

20   point of the original conic. Then each conic is changed into a cubic Bezier curve after finding its peak point, which is the point where the tangent is parallel to the chord, that can be found by using the section "Computing Points and Tangent Directions of a Conic" described below.

COMPUTING POINTS AND TANGENT DIRECTIONS OF A CONIC

Let b_0, b_1, and b_2 be the control points and w the weight of a conic. For drawing a conic or for selecting a point on a conic the following algorithm is used, which is different from the deCasteljau algorithm. This algorithm is faster than making computations using the equation of the conic. It computes a number N of points and tangent directions, the number N being a power

5 of 2 (such as N = 1024). For drawing the conic, the polygonal line connecting these points is drawn. The points are not computed in the order they are drawn, but each point has an index and the points are drawn in the order of the indices. The points p[n], n = 0, 1, 2, ..., N, start at the start point p[0] = b_0 and end at the end point p[N] = b_2. The tangent directions t[n], n = 0, 1, 2, ..., N, start at the start tangent direction t[0] = b_1 - b_0, and end at the end tangent direction t[N] =

10 b_2 - b_1. The points p[n] and tangent directions t[n], n = 1, 2, 3, ..., N – 1 are computed by a recursive function depending on an index n, a step s, control points c_0, c_1, and c_2, and a weight v (for a conic which is a portion of the original conic). This recursive function also depends on the arrays p[ ] and t[ ], where it writes the results. The first time this function is called for the index N/2, the step N/2, the control points c_0 = b_0, c_1 = b_1, and c_2= b_2, and the weight v = w, and it

15 computes the point p[N/2] and the tangent direction t[N/2]. When the function is called for the index n, the step s, the control points c_0, c_1, and c_2, and the weight u, it computes:

the center point q = 0.5(c_0 + c_2),

the factor $\lambda = \dfrac{v}{v+1}$,

20 the peak point p[n] = q + $\lambda(c_1$ – q),

and the tangent direction t[n] = c2 – c0.

After p[n] and t[n] are computed, if s is still larger than 1, the recursive function gets called twice for the index n – s/2, the step s/2,

the control points c_0, c_0 + λ (c_1 – c_0), and p[n],

and the weight $u = \sqrt{0.5(v+1)}$

(for the conic between c_0 and $p[n]$),

and for the index $n + s/2$, the step $s/2$,

the control points $p[n]$, $c_2 + \lambda (c_1 - c_2)$, and c_2,

5   and the weight $u = \sqrt{0.5(v+1)}$.

(for the conic between $p[n]$ and c_2),

The point $p[n]$ and tangent direction $t[n]$ are the point and tangent direction at the parameter $t = n/N$, and they are computed faster than using the equation of the original conic.

Conclusion

10   Although the above description is specific, it should not be considered as a limitation on the scope of the invention of the material describe herein, but only as an additional examples of the preferred embodiments. Many substitutes and variations are possible within the teachings of the invention, even for the particular example of conics. The mathematical formulas are only examples how the curves may be determined; any other suitable formulas may be used, even for

15   the particular example of conics.

For the described point-curvature curves any types curves can be used and a specific description is made for the example of cubic Bezier curves conics. Also, for the embodiments of point-tangent and point curves, in addition to the example of conics, new examples of cubic Bezier curves are described.

20   It is also shown how conics can be converted into cubic Bezier curves that are very close, and how cubic Bezier curves can be converted into conics that a fairly close, but are chosen in such a way that allows for shape improvement. The algorithms for these conversions use the peak-point curve concept.

I Claim: